

Programação de Computadores II

Ponteiro para Ponteiro

Prof. Alessandro Copetti

1

Ponteiro para Ponteiro

```
char a;  
char *b;  
char **c;  
  
a = 'z';  
b = &a;  
c = &b;  
**c = 'M';
```

• Temos dois "níveis": c aponta para b, e b aponta para a. Assim, para acessar a usando o ponteiro c, é necessário usar duas vezes o operador "*":

- uma para obter o valor de b (cujo endereço está guardado em c), e a outra para obter o valor de a, apontado por b

- Uma aplicação de ponteiro para ponteiro está em um conjunto de strings, já que cada string é um vetor, e um conjunto será um vetor desses vetores. Um **vetor de strings** seria justamente um ponteiro para um ponteiro.

2

Ponteiro para Ponteiro

```
char a;                printf("\n%c", a);  
char *b;               printf("\n%c", *b);  
char **c;              printf("\n%c", **c);  
  
a = 'z';               printf("\n%c", **c);  
b = &a;                printf("\n%c", **&b);  
c = &b;                printf("\n%c", *b);  
  
printf("\n%c", *b);  
  
O que será impresso?
```

3

Vetor de Estruturas

- Ex:
 - Aluno vet[100];

4

Vetor de Ponteiros

- Ex de vetor de ponteiros 'para estruturas':
 - Aluno *vet[100];
- Ex de vetor de ponteiros 'para tipos primitivos':
 - int *vet[100];

5

Ponteiro para Ponteiro

- Como é um vetor em que cada elemento é uma string, por exemplo, uma lista de nomes de pessoas?
- Exemplos:
- char nomes[10][80];
scanf("%79[^\n]", nomes[i]);
- char *estado[27] =
{ "AC", "AL", "AM", "AP", "BA", "CE", "DF", "ES", "GO", "MA", "MG", "MS", "MT",
 "PA", "PB", "PE", "PI", "PR", "RJ", "RN", "RO", "RR", "RS", "SC", "SE", "SP",
 "TO" };

6

Ponteiro para Ponteiro

```
char nomes[10][80];
int i=0, cont=0;
/* Ou a inicialização por alocação
dinâmica:
char **nomes;
scanf("%d",&numNomes);
nomes = (char **)
    malloc(numNomes*sizeof(char
    *));
for (i=0; i<numNomes; i++)
    nomes[i]=(char *)
        malloc(80*sizeof(char)); */

while (1) {
    scanf("%79[^\n]", nomes[cont]);
    if (strcmp(nomes[cont],"fim")==0)
        break;
    cont++;
}
for (i=0; i < cont; i++)
    printf("\n%s", nomes[i]);
//teste:
printf("\n%c",nomes[0][2]);
```

7

Ponteiro para Ponteiro

```
nomes = (char **) malloc(numNomes*sizeof(char *));
//equivalente ao número de linhas

for (i=0; i<numNomes; i++)
    nomes[i]=(char *) malloc(80*sizeof(char));
//equivalente ao número de colunas
```

8